

<<编程珠玑>>

图书基本信息

书名：<<编程珠玑>>

13位ISBN编号：9787115179289

10位ISBN编号：711517928X

出版时间：2008-10

出版时间：人民邮电出版社

作者：Jon Bentley

页数：228

译者：黄倩,钱丽艳

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<编程珠玑>>

前言

本书作者Jon Bentley是美国著名的程序员和计算机科学家，他于20世纪70年代前后在很有影响力的《ACM通讯》（Communications of the ACM）上以专栏的形式连续发表了一系列短文，成功地总结和提炼了自己在长期的计算机程序设计实践中积累下来的宝贵经验。

这些短文充满了真知灼见，而且文笔生动、可读性强，对于提高职业程序员的专业技能很有帮助，因此该专栏大受读者欢迎，成为当时该学术期刊的王牌栏目之一。

可以想象当时的情形颇似早年金庸先生在《明报》上连载其武侠小说的盛况。

后来在ACM的鼓励下，作者经过仔细修订和补充整理，对各篇文章的先后次序做了精心编排，分别在1986年和1988年结集出版了Programming Pearls（《编程珠玑》）和More Programming Pearls（《编程珠玑》）这两本书，二者均成为该领域的名著。

《编程珠玑（第2版）》在2000年问世，书中的例子都改用C语言书写，并多处提到如何用C++和Java中的类来实现。

《编程珠玑》虽未再版，例子多以Awk语言写成，但其语法与C相近，容易看懂。

作者博览群书，旁征博引，无论是计算机专业的专业名著，如《计算机程序设计艺术》，还是普通的科普名著，如《啊哈！

灵机一动》，都在作者笔下信手拈来、娓娓道出，更不用说随处可见的作者自己的真知灼见了。

如果说《计算机程序设计艺术》这样的巨著代表了程序员们使用的“坦克和大炮”一类的重型武器，这两本书则在某种程度上类似于鲁迅先生所说的“匕首与投枪”一类的轻型武器，更能满足职业程序员的日常需要。

或者说前者是武侠小说中提高内力修为的根本秘籍，后者是点拨临阵招数的速成宝典，二者同样都是克敌制胜的法宝，缺一不可。

在无止境地追求精湛技艺这一点上，程序员、数学家和武侠们其实是相通的。

在美国，这两本书不仅被用作大学低年级数据结构与算法课程的教材，还用作高年级算法课程的辅助教材。

例如，美国著名大学麻省理工学院的电气工程与计算机科学开放式核心课程算法导论就将这两本书列为推荐读物。

这两本书覆盖了大学算法课程和数据结构课程的大部分内容，但是与普通教材的侧重点又不一样，不强调单纯从数学上来进行分析的技巧，而是强调结合实际问题来进行分析、应用和实现的技巧，因此可作为大学计算机专业的算法、数据结构、软件工程等课程的教师参考用书和优秀课外读物。

书中有许多真实的历史案例和许多极好的练习题以及部分练习题的提示与解答，非常适合自学。

正如作者所建议的那样，阅读这两本书时，读者需要备有纸和笔，最好还有一台计算机在手边，边读边想、边想边做，这样才能将阅读这两本书的收益最大化。

<<编程珠玑>>

内容概要

本书是计算机科学方面的经典名著。

书的内容围绕程序设计人员面对的一系列实际问题展开。

作者Jon Bentley 以其独有的洞察力和创造力，引导读者理解这些问题并学会解决方法，而这些正是程序员实际编程生涯中至关重要的。

本书的特色是通过一些精心设计的有趣而又颇具指导意义的程序，对实用程序设计技巧及基本设计原则进行了透彻而睿智的描述，为复杂的编程问题提供了清晰而完备的解决思路。

本书对各个层次的程序员都具有很高的阅读价值。

<<编程珠玑>>

作者简介

Jon Bentley, 世界著名计算机科学家, 被誉为影响算法发展的十位大师之一。

他先后任职于卡内基—梅隆大学(1976—1982)、贝尔实验室(1982—2001)和Avaya实验室(2001年至今)。

在卡内基—梅隆大学担任教授期间, 他培养了包括Tcl语言设计者John Ousterhout、Java语言设计者James Gosling、《算法导论》作者之一Charles Leiserson在内的许多计算机科学大家。

2004年荣获Dr. Dobb's程序设计卓越奖。

钱丽艳, 北京大学信息科学技术学院基础实验教学研究所软件实验室主任、工程师, 毕业于国防科技大学。

<<编程珠玑>>

书籍目录

第一部分 基础	第1章 开篇	1.1 一次友好的对话	1.2 准确的问题描述	1.3 程序设计
	1.4 实现概要	1.5 原理	1.6 习题	1.7 深入阅读
算法	2.1 三个问题	2.2 无处不在的二分搜索	2.3 基本操作的威力	2.4 排序
	2.5 原理	2.6 习题	2.7 深入阅读	2.8 变位词程序的实现(边栏)
第3章 数据决定程序结构	3.1 一个调查程序	3.2 格式信函编程	3.3 一组示例	
	3.4 结构化数据	3.5 用于特殊数据的强大工具	3.6 原理	3.7 习题
	3.8 深入阅读			
第4章 编写正确的程序	4.1 二分搜索的挑战	4.2 编写程序	4.3 理解程序	
	4.4 原理	4.5 程序验证的角色	4.6 习题	4.7 深入阅读
第5章 编程小事	5.1 从伪代码到C程序	5.2 测试工具	5.3 断言的艺术	5.4 自动测试
	5.5 计时	5.6 完整的程序	5.7 原理	5.8 习题
	5.9 深入阅读			
5.10 调试(边栏)				
第二部分 性能	第6章 程序性能分析	6.1 实例研究	6.2 设计	
层面	6.3 原理	6.4 习题	6.5 深入阅读	第7章 粗略估算
	7.1 基本技巧			
	7.2 性能估计	7.3 安全系数	7.4 Little定律	7.5 原理
	7.6 习题			
7.7 深入阅读	7.8 日常生活中的速算(边栏)			
第8章 算法设计技术	8.1 问题及简单算法	8.2 两个平方算法	8.3 分治算法	8.4 扫描算法
	8.5 实际运行时间	8.6 原理	8.7 习题	8.8 深入阅读
第9章 代码调优	9.1 典型的故事	9.2 急救方案集锦	9.3 大手术——二分搜索	9.4 原理
	9.5 习题	9.6 深入阅读		
第10章 节省空间	10.1 关键在于简单	10.2 示例问题	10.3 数据空间技术	10.4 代码空间技术
	10.5 原理	10.6 习题	10.7 深入阅读	10.8 巨大的节省(边栏)
第三部分 应用	第11章 排序	11.1 插入排序	11.2 一种简单的快速排序	11.3 更好的几种快速排序
	11.4 原理	11.5 习题	11.6 深入阅读	
第12章 取样问题	12.1 问题	12.2 一种解决方案	12.3 设计空间	12.4 原理
	12.5 习题	12.6 深入阅读		
第13章 搜索	13.1 接口	13.2 线性结构	13.3 二分搜索树	13.4 用于整数的结构
	13.5 原理	13.6 习题		
13.7 深入阅读	13.8 一个实际搜索问题(边栏)			
第14章 堆	14.1 数据结构	14.2 两个关键函数	14.3 优先级队列	14.4 一种排序算法
	14.5 原理	14.6 习题		
14.7 深入阅读				
第15章 字符串	15.1 单词	15.2 短语	15.3 生成文本	15.4 原理
	15.5 习题	15.6 深入阅读		
第1版跋	第2版跋	附录A 算法分类	附录B 估算测试	附录C 时空开销模型
附录D 代码调优法则	附录E 用于搜索的C++类	部分习题提示	部分习题答案	索引

<<编程珠玑>>

章节摘录

插图：第一部分 基础第1章 开篇1.2 准确的问题描述对程序员来说，这些需求加起来就是：“如何给磁盘文件排序？”

在试图解决这个问题之前，先将已知条件组织成一种更客观、更易用的形式。

输入：一个最多包含 n 个正整数的文件，每个数都小于 n ，其中 $n=107$ 。

如果在输入文件中有任何整数重复出现就是致命错误。

没有其他数据与该整数相关联。

输出：按升序排列的输入整数的列表。

约束：最多有（大约）1MB的内存空间可用，有充足的磁盘存储空间可用。

运行时间最多几分钟，运行时间为10秒就不需要进一步优化了。

请花上一分钟思考一下该问题的规范说明。

现在你打算给程序员什么样的建议呢？

1.3 程序设计显而易见的方法是以一般的基于磁盘的归并排序程序为起点，但是要对其进行调整，因为我们对整数进行排序。

这样就可以将原来的200行程序减少为几十行，同时也使得程序运行得更快，但是完成程序并使之运行可能仍然需要几天的时间。

另一种解决方案更多地利用了该排序问题的特殊性。

如果每个号码都使用7个字节来存储，那么在可用的1MB存储空间里大约可以存143 000个号码。

如果每个号码都使用32位整数来表示的话，在1MB存储空间里就可以存储250000个号码。

因此，可以使用遍历输入文件40趟的程序来完成排序。

在第一趟遍历中，将0至249999之间的任何整数都读入内存，并对这（最多）250000个整数进行排序，然后写到输出文件中。

第二趟遍历排序250000至499999之间的整数，依此类推，到第40趟遍历的时候对9750000至9999999之间的整数进行排序。

对内存中的排序来说，快速排序会相当高效，而且仅仅需要20行代码。

于是，整个程序就可以通过一两页纸的代码实现。

该程序拥有所期望的特性——不必考虑使用中间磁盘文件；不幸的是，为此所付出的代价是要读取输入文件40次。

1.5 原理那个程序员打电话把他的问题告诉我，然后我们花了大约一刻钟时间明确了问题所在，并找到了位图解决方案。

他花了几个小时来实现这个几十行代码的程序。

该程序远远优于我们在电话刚开始时所担心的需要花费一周时间编写的几百行代码的那个程序。

而且程序执行得很快：磁盘上的归并排序可能需要许多分钟的时间，该程序所需的时间只比读取输入和写入输出所需的时间多一点点——大约10秒钟。

答案3包含了对完成该任务的几种不同程序的计时细节。

从这些事实中可以总结出该实例研究所得到的第一个结论：对小问题的仔细分析有时可以得到明显的实际益处。

在该实例中，几分钟的仔细研究可以大幅削减代码的长度、程序员时间和程序运行时间。

Chuck Yeager将军（第一个超音速飞行的人）赞扬一架飞机的机械系统时用的词是“结构简单、部件很少、易于维护、非常坚固”，该程序拥有同样的属性。

然而，当规范说明的某些因素发生改变时，该程序的特殊结构将很难修改。

除了需要精巧的编程以外，该实例阐明了如下一般原理。

正确的问题。

明确问题，这场战役就成功了90%——我很庆幸程序员没有满足于我给出的第一个程序。

一旦正确理解了问题，习题10、习题11和习题12的答案都会很优雅。

在查看提示和答案以前，请努力思考这些问题。

<<编程珠玑>>

位图数据结构。

该数据结构描述了一个有限定义域内的稠密集合，其中的每一个元素最多出现一次并且没有其他任何数据与该元素相关联。

即使这些条件没有完全满足（例如，存在重复元素或额外的数据），也可以用有限定义域内的作为一个表项更复杂的表格的索引，见习题6和习题8。

多趟算法。

这些算法多趟读入其输入数据，每次完成一步。

在1.3节已经见到了一个40趟算法，习题5鼓励读者去完成一个两趟算法。

时间—空间折中与双赢。

编程文献和理论中充斥着时间—空间的折中：通过使用更多的时间，可以减少程序所需的空间。

例如，答案5中的两趟算法让程序运行时间加倍从而使空间减半。

但我的经验常常是这样的：减少程序的空间需求也会减少其运行时间。

空间上高效的位图结构显著地减少了排序的运行时间。

空间需求的减少之所以会导致运行时间的减少，有两个原因：需要处理的数据变少了，意味着处理这些数据所需的时间也变少了；同时将这些数据保存在内存中而不是磁盘上，进一步避免了磁盘访问的时间。

当然了，只有在原始的设计远非最佳方案时，才有可能时空双赢。

简单的设计。

Antoine de Saint—Exupery是法国作家兼飞机设计师，他曾经说过：“设计者确定其设计已经达到了完美的标准不是不能再增加任何东西，而是不能再减少任何东西。

”更多的程序员应该使用该标准来检验自己完成的程序。

简单的程序通常比具有相同功能的复杂程序更可靠、更安全、更健壮、更高效，而且易于实现和维护

。

<<编程珠玑>>

后记

本书作者Jon Bentley是美国著名的程序员和计算机科学家，他于20世纪70年代前后在很有影响力的《ACM通讯》（Communications of the ACM）上以专栏的形式连续发表了一系列短文，成功地总结和提炼了自己在长期的计算机程序设计实践中积累下来的宝贵经验。

这些短文充满了真知灼见，而且文笔生动、可读性强，对于提高职业程序员的专业技能很有帮助，因此该专栏大受读者欢迎，成为当时该学术期刊的王牌栏目之一。

可以想象当时的情形颇似早年金庸先生在《明报》上连载其武侠小说的盛况。

后来在ACM的鼓励下，作者经过仔细修订和补充整理，对各篇文章的先后次序做了精心编排，分别在1986年和1988年结集出版了Programming Pearls（《编程珠玑》）和More Programming Pearls（《编程珠玑 》）这两本书，二者均成为该领域的名著。

《编程珠玑（第2版）》在2000年问世，书中的例子都改用C语言书写，并多处提到如何用C++和Java中的类来实现。

《编程珠玑 》虽未再版，例子多以Awk语言写成，但其语法与C相近，容易看懂。

作者博览群书，旁征博引，无论是计算机专业的专业名著，如《计算机程序设计艺术》，还是普通的科普名著，如《啊哈！

灵机一动》，都在作者笔下信手拈来、娓娓道出，更不用说随处可见的作者自己的真知灼见了。

如果说《计算机程序设计艺术》这样的巨著代表了程序员们使用的“坦克和大炮”一类的重型武器，这两本书则在某种程度上类似于鲁迅先生所说的“匕首与投枪”一类的轻型武器，更能满足职业程序员的日常需要。

或者说前者是武侠小说中提高内力修为的根本秘籍，后者是点拨临阵招数的速成宝典，二者同样都是克敌制胜的法宝，缺一不可。

在无止境地追求精湛技艺这一点上，程序员、数学家和武侠们其实是相通的。

在美国，这两本书不仅被用作大学低年级数据结构与算法课程的教材，还用作高年级算法课程的辅助教材。

例如，美国著名大学麻省理工学院的电气工程与计算机科学开放式核心课程算法导论就将这两本书列为推荐读物。

这两本书覆盖了大学算法课程和数据结构课程的大部分内容，但是与普通教材的侧重点又不一样，不强调单纯从数学上来进行分析的技巧，而是强调结合实际问题来进行分析、应用和实现的技巧，因此可作为大学计算机专业的算法、数据结构、软件工程等课程的教师参考用书和优秀课外读物。

书中有许多真实的历史案例和许多极好的练习题以及部分练习题的提示与解答，非常适合自学。

正如作者所建议的那样，阅读这两本书时，读者需要备有纸和笔，最好还有一台计算机在手边，边读边想、边想边做，这样才能将阅读这两本书的收益最大化。

<<编程珠玑>>

编辑推荐

多年以来，当程序员们推选出最心爱的计算机图书时，《编程珠玑》总是位于前列。正如自然界里珍珠出自细沙对牡蛎的磨砺。

计算机科学大师Jon Bentley以其独有的洞察力和创造力，从磨砺程序员的实际问题中凝结出一篇篇不朽的编程“珠玑”，成为世界计算机界名刊《ACM通讯》历史上最受欢迎的专栏，最终结集为两部不朽的计算机科学经典名著，影响和激励着一代又一代程序员和计算机科学工作者。

本书为第一卷，主要讨论计算机科学中最本质的问题：如何正确选择和高效地实现算法。

在书中，作者选取许多具有典型意义的复杂编程和算法问题，生动描绘了历史上众大师们在探索解决方案中发生的轶事、走过的弯路和不断精益求精的历程，引导读者像真正的程序员和软件工程师那样富于创新性地思考，并透彻阐述和总结了许多独特而精妙的设计原则、思考和解决问题的方法以及实用程序设计技巧。

解决方案的代码均以C / C++语言编写，不仅有趣，而且有很大的实战示范意义。

每章后所附习题极具挑战性和启发性，书末给出了简洁的解答。

<<编程珠玑>>

名人推荐

“《编程珠玑》第1版是对我职业生涯早期影响最大的书之一，其中的许多真知灼见多年之后仍然使我受益匪浅。

Jon在第2版中对素材进行了大量更新，许多新内容让我耳目一新。

” ——Steve McConnell，软件工程大师，IEEE Software前主编。

《代码大全》作者“对每一位遇到的程序员，我都会毫不迟疑地建议他阅读并不断重读这部经典之作

。”
” ——Slashdot

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>